

Fast Calculation of Multiobjective Probability of Improvement and Expected Improvement Criteria for Pareto Optimization

Ivo Couckuyt · Dirk Deschrijver · Tom Dhaene

the date of receipt and acceptance should be inserted later

Abstract The use of Surrogate Based Optimization (SBO) is widely spread in engineering design to reduce the number of computational expensive simulations. However, “real-world” problems often consist of multiple, conflicting objectives leading to a set of competitive solutions (the Pareto front). The objectives are often aggregated into a single cost function to reduce the computational cost, though a better approach is to use multiobjective optimization methods to directly identify a set of Pareto-optimal solutions, which can be used by the designer to make more efficient design decisions (instead of weighting and aggregating the costs upfront). Most of the work in multiobjective optimization is focused on MultiObjective Evolutionary Algorithms (MOEAs). While MOEAs are well-suited to handle large, intractable design spaces, they typically require thousands of expensive simulations, which is prohibitively expensive for the problems under study. Therefore, the use of surrogate models in multiobjective optimization, denoted as MultiObjective Surrogate-Based Optimization (MOSBO), may prove to be even more worthwhile than SBO methods to expedite the optimization of computational expensive systems. In this paper, the authors propose the Efficient Multiobjective Optimization (EMO) algorithm which uses Kriging models and multiobjective versions of the Probability of Improvement (PoI) and Expected Improvement (EI) criteria to identify the Pareto front with a minimal number of expensive simulations. The EMO algorithm is applied on multiple standard benchmark problems and compared against the well-known NSGA-II, SPEA2 and SMS-EMOA multiobjective optimization methods.

Keywords multiobjective optimization · expected improvement · probability of improvement · hypervolume · Kriging · Gaussian Process

1 Introduction

Surrogate modeling techniques, also known as metamodeling, are becoming rapidly popular in the engineering community to speed up complex, computational expensive

Ivo Couckuyt
Department of Information Technology (INTEC), Ghent University-iMinds, Ghent, Belgium
E-mail: ivo.couckuyt@ugent.be

design problems [37, 22]. Surrogate models, or metamodels, are mathematical approximation models that mimic the behavior of computational expensive simulation codes such as mechanical or electrical finite element simulations, or computational fluid dynamic simulations. This paper deals with the use of surrogate models for expediting the optimization of time-consuming (black-box) problems of a deterministic nature, in contrast to stochastic simulation.

While several types of surrogate modeling uses can be distinguished, this work is concerned with the integration of surrogate models into the optimization process, often denoted by Surrogate Based Optimization (SBO) or Metamodel-Assisted Optimization (MAO). SBO methods typically generate surrogate models on the fly that are only accurate in certain regions of the input space, e.g., around potentially optimal regions. The generated surrogate models can then be used to intelligently guide the optimization process to the global optimum.

The focus of this work is the global SBO method based on the Probability of Improvement (PoI) and Expected Improvement (EI), popularized by Jones et al. [25]. These “*statistical criteria*” guide the selection of new data points in such a way that the objective function is optimized, while minimizing the number of expensive simulations. The advantage of EI and PoI is that, besides the prediction (mean), the uncertainty (variance) of the surrogate model is taken into account as well, providing a balance between exploration¹ and exploitation². Most often EI or PoI is used in conjunction with the Kriging surrogate model (Gaussian Processes) [27] which provides by construction a prediction of the mean as well as the variance, but other surrogate models are also possible, such as Radial Basis Functions (RBF), Support Vector Regression (SVR) [13], etc.

The single-objective SBO problem is well described in literature, however, most (if not all) “*real-world*” problems actually consists of multiple, conflicting objectives leading to a set of Pareto-optimal solutions. Often the objectives are aggregated into a single cost function, e.g., using a weighted sum, that can be optimized by standard optimization techniques. Subsequently, by repeating this process many times using varying starting conditions, e.g., different set of weights, several solutions on the Pareto front can be found. On the other hand, a multiobjective optimization method can optimize the different objective functions simultaneously, and try to find the Pareto front in just a single run. Examples of such methods are primarily the MultiObjective Evolutionary Algorithms (MOEAs), e.g., the “Non-dominated Sorting Genetic Algorithm II” (NSGA-II; [14]), the “Strength Pareto Evolutionary Algorithm 2” (SPEA2; [44]) and the “S-Metric Selection Evolutionary MultiObjective Algorithm” (SMS-EMOA; [5]).

Unfortunately, MOEAs typically require a massive amount of function evaluations, which is infeasible for computational expensive simulators. Hence, it is vital to economize on the number of function evaluations, e.g., by using surrogate models. MultiObjective Surrogate-based Optimization (MOSBO) methods only appeared quite recently in literature. Most work is focused on integrating surrogate models in MOEAs [41]. Gaspar et al. [21] use neural networks to either approximate the fitness function or as a local approximation technique to generate search points more efficiently. Voutchkov et al. [35] apply the NSGA-II algorithm to Kriging models instead of the expensive sim-

¹ Improving the overall accuracy of the surrogate model (space-filling).

² Enhancing the accuracy of the surrogate model solely in the region of the (current) optimum.

ulator. For an overview of available techniques and approaches the reader is referred to [30,42].

While the PoI and EI approach is well-developed and used for single-objective SBO, its use in MOSBO is not well spread. Single-objective versions of EI and PoI are utilized by Knowles et al. [28,29] to solve MOSBO problems. This approach, known as ParEGO, uses Kriging and EI to optimize a weighted sum of objective functions. By randomizing the weights every iteration several solutions along the Pareto front can be identified. More recently, Keane [26] proposed multiobjective versions of PoI and Euclidean distance-based EI. At the same time Emmerich et al. [17] proposed the hypervolume-based EI criterion. Similarly to a weighted sum, the multiobjective versions of EI and PoI aggregate information from the surrogate models into a single cost function, balancing between exploration¹ and exploitation³. Unfortunately, only formulae for two objective functions are given by Keane as the statistical criteria become rather cumbersome and complex for a higher number of objective functions. Similarly, while Emmerich et al. [16] describe formulae for an arbitrary number of dimensions for the hypervolume-based EI, the computation cost increases at least exponentially with the number of objectives and, hence, has only been applied to two objectives.

The key contribution of this paper is the Efficient Multiobjective Optimization (EMO) algorithm which is a much more efficient method of evaluating multiobjective versions of the PoI and EI criteria for multiobjective optimization problems. In fact, the problem at hand is similar to calculating the hypervolume (a Pareto set quality estimator) [45] as will be shown below and, hence, hypervolume algorithms can be adapted to aid in the evaluation of the statistical criteria. Moreover, a new statistical criterion is proposed, based on the hypervolume-based EI, which is significantly cheaper to compute while still delivering promising results.

In section 2 the Kriging surrogate model is briefly discussed. In section 3, an overview of the EMO algorithm is given, including general expressions for the PoI and several variants of EI. Subsequently, a fundamental part needed for the calculation of the statistical criteria is discussed in section 3.4. Afterwards, in section 4 the EMO algorithm is tested on several functions from the DTLZ benchmark suite [15]. Lastly, in section 5 conclusions and future work are discussed.

2 Kriging

Kriging is a popular surrogate model to approximate deterministic noise-free data, and has proven to be very useful for tasks such as optimization [25], design space exploration, visualization, prototyping, and sensitivity analysis [37].

A thorough mathematical treatment of Kriging is given in [33,19]. Basically, Kriging is a two-step process: first a regression function $h(\mathbf{x})$ is constructed, and, subsequently, a centered Gaussian process Z with variance σ^2 and a correlation matrix Ψ is constructed through the residuals.

$$Y(\mathbf{x}) = h(\mathbf{x}) + Z(\mathbf{x}). \quad (1)$$

Consider a set of n samples, $(\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$ in d dimensions (see Equation 2) and associated function values, $\mathbf{y} = (y_1, \dots, y_n)^\top$, where $(\cdot)^\top$ is the transpose of a vector or matrix.

³ Improving or augmenting the Pareto front.

$$X = \begin{pmatrix} \mathbf{x}_1, \dots, \mathbf{x}_n \end{pmatrix}^\top = \begin{pmatrix} x_{1,1} & \dots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \dots & x_{n,d} \end{pmatrix} \quad (2)$$

Essentially, the regression part is encoded in the $n \times p$ model matrix F using basis functions $b_i(\mathbf{x})$ for $i = 1 \dots p$,

$$F = \begin{pmatrix} b_1(\mathbf{x}_1) & b_2(\mathbf{x}_1) & \dots & b_p(\mathbf{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ b_1(\mathbf{x}_n) & b_2(\mathbf{x}_n) & \dots & b_p(\mathbf{x}_n) \end{pmatrix},$$

while the stochastic process is mainly defined by the $n \times n$ correlation matrix Ψ ,

$$\Psi = \begin{pmatrix} \psi(\mathbf{x}_1, \mathbf{x}_1) & \dots & \psi(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ \psi(\mathbf{x}_n, \mathbf{x}_1) & \dots & \psi(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix},$$

where $\psi(\cdot, \cdot)$ is the correlation function. $\psi(\cdot, \cdot)$ is parameterized by a set of hyperparameters θ . The choice of correlation function is crucial to obtain good accuracy. This paper focuses on using the Matérn correlation function [34], with $\nu = 3/2$,

$$\psi(\mathbf{x}, \mathbf{x}')^{\text{Matérn}}_{\nu=3/2} = \left(1 + \sqrt{3}l\right) \exp\left(-\sqrt{3}l\right),$$

with $l = \sqrt{\sum_{i=1}^d \theta_i (x_i - x'_i)^2}$. In addition, the popular Gaussian correlation function is also used,

$$\psi(\mathbf{x}, \mathbf{x}')^{\text{Gauss}} = \exp\left(-\sum_{i=1}^d \theta_i |x_i - x'_i|^2\right).$$

The hyperparameters θ are identified by Maximum Likelihood Estimation (MLE). In particular, the negative concentrated log-likelihood is minimized,

$$\underset{\theta}{\operatorname{argmin}} -\frac{n}{2} \ln(\sigma^2) - \frac{1}{2} \ln(|\Psi|),$$

where $\sigma^2 = \frac{1}{n}(\mathbf{y} - F\alpha)^\top \Psi^{-1}(\mathbf{y} - F\alpha)$. Subsequently, the prediction mean and prediction variance of Kriging are derived, respectively, as,

$$\mu(\mathbf{x}) = M\alpha + r(\mathbf{x}) \cdot \Psi^{-1} \cdot (\mathbf{y} - F\alpha), \quad (3)$$

$$s^2(\mathbf{x}) = \sigma^2 \left(1 - r(\mathbf{x})\Psi^{-1}r(\mathbf{x})^\top + \frac{(1 - F^\top \Psi^{-1}r(\mathbf{x})^\top)}{F^\top \Psi^{-1}F}\right), \quad (4)$$

where $M = \begin{pmatrix} b_1(\mathbf{x}) & b_2(\mathbf{x}) & \dots & b_p(\mathbf{x}) \end{pmatrix}$ is the model matrix of the predicting point \mathbf{x} , α is a $p \times 1$ vector denoting the coefficients of the regression function, determined by Generalized Least Squares (GLS), and $r(\mathbf{x})$ is an $1 \times n$ vector of correlations between the point \mathbf{x} and the samples X .

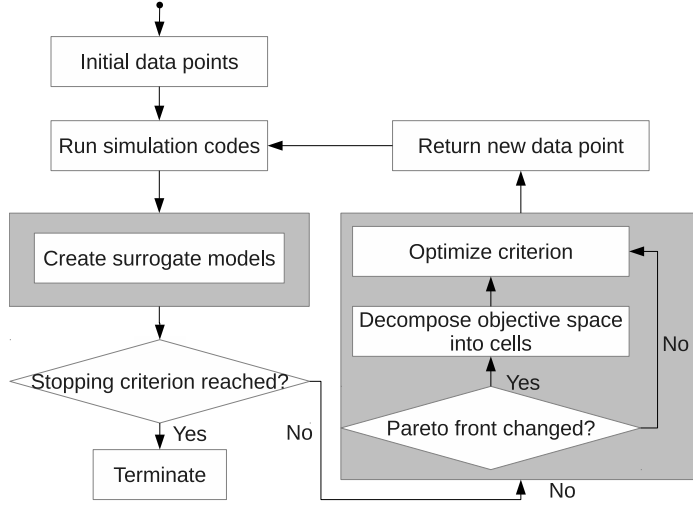


Fig. 1: Flow chart of the Efficient Multiobjective Optimization (EMO) algorithm.

3 Efficient Multiobjective Optimization (EMO)

3.1 Overview

A flow chart of the EMO algorithm is shown in Figure 1. First an initial set of points X is generated and evaluated on the expensive objective functions $f_j(\mathbf{x})$, for $j = 1 \dots m$. Each objective function $f_j(\mathbf{x})$ is then approximated by a Kriging model. Based on the Kriging models useful criteria can be constructed that help in identifying Pareto-optimal solutions. After selecting a new point it is evaluated on the expensive objective functions $f_j(\mathbf{x})$, the Kriging models are updated with this new information and this process is repeated in an iterative fashion until some stopping criterion is met.

Of particular interest are the Probability of Improvement (PoI) and Expected Improvement (EI) statistical criteria which are widely used for single-objective optimization [24, 10]. Hence, it may be useful to extend the concept of the PoI and EI directly to multiobjective optimization. Multiobjective versions of the PoI and EI are defined for an arbitrary number of objective functions in sections 3.2 and 3.3.

For ease of notation in the forthcoming sections, the output of all the Kriging models can be considered as mutually independent Gaussian random variables $Y_j(\mathbf{x})$,

$$Y_j(\mathbf{x}) \sim \mathcal{N}(\mu_j(\mathbf{x}), s_j^2(\mathbf{x})) \text{ for } j = 1 \dots m. \quad (5)$$

The associated probability density function ϕ_j and cumulative distribution function Φ_j of $Y_j(\mathbf{x})$ are compactly denoted as,

$$\phi_j[y_j] \triangleq \phi_j[y_j; \mu_j(\mathbf{x}), s_j^2(\mathbf{x})], \quad (6)$$

$$\Phi_j[y_j] \triangleq \Phi_j[y_j; \mu_j(\mathbf{x}), s_j^2(\mathbf{x})]. \quad (7)$$

Given a set of n points X as in (2), a Pareto set \mathcal{P} can be constructed that comprises $v \leq n$ Pareto-optimal (non-dominated) solutions,

$$\mathcal{P} = \{\mathbf{f}(\mathbf{x}_1^*), \dots, \mathbf{f}(\mathbf{x}_v^*)\}. \quad (8)$$

Each solution $\mathbf{f}(\mathbf{x}_i^*)$ is a vector that contains the objective function values for an associated input point $\mathbf{x}_i^* \in X$, for $i = 1 \dots v$,

$$\mathbf{f}(\mathbf{x}_i^*) = (f_1(\mathbf{x}_i^*), \dots, f_m(\mathbf{x}_i^*)). \quad (9)$$

3.2 Probability of Improvement (PoI)

Evaluating the probability that the objective function values of a new input point \mathbf{x} are located inside a well-defined region A in the objective space requires a multidimensional integration over that region. Naturally, several variants of the multiobjective PoI can be constructed as the concept of improvement is ambiguously defined in the context of multiobjective optimization. This is reflected in the selection of the integration region A , e.g., A can be the non-dominated part of the objective space or A can be the region in the objective space that solely extends (and does not dominate) the Pareto set, etc. In general, the probability that a new input point \mathbf{x} yields improvement over the Pareto set \mathcal{P} is denoted by the PoI $P[I]$,

$$P[I] = \int_{\mathbf{y} \in A} \prod_{j=1}^m \phi_j[y_j] dy_j. \quad (10)$$

To evaluate Equation (10), the integration area A can be decomposed into q (hyper-)rectangular cells, which yields a finite summation of contributing terms, see Figure 2a. The lower and upper bound $[l^k, u^k]$ of each cell, for $k = 1 \dots q$, will be computed in section 3.4.

$$P[I] = \sum_{k=1}^q \pm \prod_{j=1}^m (\Phi_j[u_j^k] - \Phi_j[l_j^k]). \quad (11)$$

While the cells can be chosen to disjointedly cover the integration area A , the algorithm described in section 3.4 decomposes the region A in overlapping cells. In this case, cells may negate the overlapping contribution of other cells by subtraction, denoted by the \pm symbol in Equation (11).

3.3 Expected Improvement (EI)

While the PoI criterion is already quite useful and insensitive to the scaling of the objective functions, it does not, necessarily, encourage the generation of a uniform Pareto set. The EI quantifies the amount of improvement using an improvement function $I(\mathbf{y}, \mathcal{P})$ and, thus, prefers solutions that are lying farther from existing members of the Pareto set. The EI integral is defined as,

$$E[I] = \int_{\mathbf{y} \in A} I(\mathbf{y}, \mathcal{P}) \prod_{j=1}^m \phi_j[y_j] dy_j. \quad (12)$$

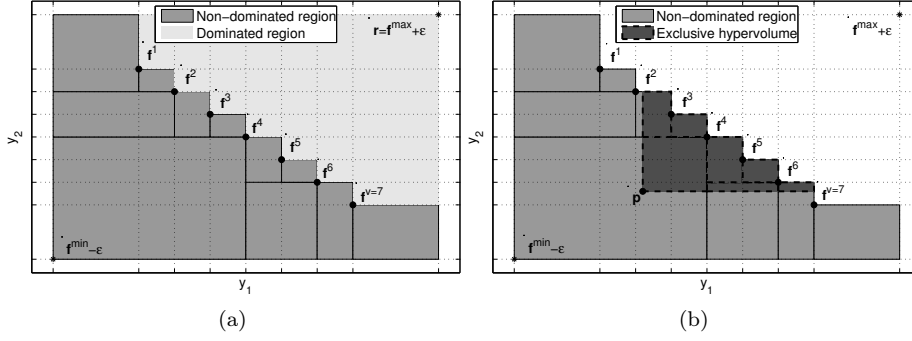


Fig. 2: Illustration of a Pareto set of two objective functions. The dots represent the Pareto points f^i , for $i = 1 \dots v$, while f^{min} and f^{max} denote the ideal and anti-ideal point, respectively. a) The dark and light shaded regions denote the non-dominated and dominated region, respectively. The volume of the latter region is the hypervolume indicator, bounded by a reference point $r = f^{max} + \epsilon$. b) The integration area A of the hypervolume-based PoI corresponds to the (light and dark) shaded region which is decomposed into cells by a binary partitioning procedure. The exclusive hypervolume of a point y relative to the Pareto set can be computed from existing cells and corresponds to the dark shaded region.

In contrast to the various types of the PoI criteria (e.g., dominating 4 Pareto points or extending the Pareto set as in section 3.2), it arguably makes more sense to only integrate the EI criteria over the region A corresponding to the non-dominated part of the objective space. The improvement function will automatically prefer new points that dominate the most points within the Pareto set \mathcal{P} (the largest improvement). When no such points are found, the improvement function encourages the selection of points that extend the Pareto set \mathcal{P} in an uniform way. Consequently, the design of the improvement function for the EI is crucial in identifying an optimal and uniform Pareto set. A good theoretical overview of different types of EI is given by [36], including work on scalar improvement functions [26,16] as well as using the single-objective EI in a multiobjective setting [28,23]. Below we focus on evaluating the Euclidean distance-based EI [26] as well as the hypervolume-based EI [17,16] efficiently for many objectives. In addition, a simplified version of the hypervolume-based EI is proposed that is significantly cheaper to compute.

3.3.1 Hypervolume-based improvement function

The hypervolume metric (or \mathcal{S} -metric) [45] is widely used in multiobjective optimization to assess the quality of a Pareto set or to drive multiobjective optimization algorithms [5]. The hypervolume indicator $\mathcal{H}(\mathcal{P})$ denotes the volume of the region dominated by the Pareto set \mathcal{P} , bounded by a reference point r which needs to be dominated by all points of the Pareto set, see Figure 2a. Larger values of the hypervolume indicates better Pareto sets. Moreover, the exclusive hypervolume (or hypervolume contribution, see Figure 2b) of a Pareto set \mathcal{P} relative to a point p is defined as,

$$\mathcal{H}_{exc}(\mathbf{p}, \mathcal{P}) = \mathcal{H}(\mathcal{P} \cup \{\mathbf{p}\}) - \mathcal{H}(\mathcal{P}). \quad (13)$$

\mathcal{H}_{exc} measures the contribution (or improvement) of the point \mathbf{p} to the Pareto set \mathcal{P} and, hence, can also be used to define a scalar improvement function, namely,

$$I(\mathbf{p}, \mathcal{P}) = \begin{cases} \mathcal{H}_{exc}(\mathbf{p}, \mathcal{P}) & \text{if } \mathbf{p} \text{ is not dominated by } \mathcal{P} \\ 0 & \text{otherwise} \end{cases}. \quad (14)$$

Subsequently, the integral of the hypervolume-based EI is,

$$E_{hv}[I] = \int_{\mathbf{y} \in A} I(\mathbf{y}, \mathcal{P}) \prod_{j=1}^m \phi_j[y_j] dy_j, \quad (15)$$

where A is the non-dominated region (bounded by the reference point \mathbf{r}).

Initially it was suggested to approximate the hypervolume-based EI using Monte Carlo techniques [17]. Recently, Emmerich et al. [16] proposed a method to calculate it exactly for an arbitrary number of dimensions by decomposing the non-dominated region into a set of cells as is also done in this work. Unfortunately, in [16] the proposed mathematical expressions assume that the non-dominated region is decomposed into a uniform grid of cells based on the Pareto set, see the cells bounded by the dashed lines in Figure 2a. Hence, the number of cells required to evaluate the criterion scales at least exponentially with the number of Pareto points and objectives. Moreover, for each cell a separate hypervolume calculation needs to be done and, hence, it is infeasible to apply the method for three objectives or higher. This work develops a new mathematical expression for the hypervolume-based EI that alleviates some of its computational complexities by decomposing the non-dominated region into a much smaller set of cells as well as removing the separate hypervolume calculations.

In contrast to other statistical criteria, the proposed expressions for the hypervolume-based EI requires the non-dominated region to be covered by a set of *disjoint* cells, then the hypervolume-based EI can be written in closed form as,

$$E_{hv}[I] = \sum_{k=1}^q IC_k \quad (16)$$

where IC_k denotes the improvement contribution of cell k , namely,

$$\begin{aligned} IC_k &= \int_{l_1^k}^{u_1^k} \dots \int_{l_m^k}^{u_m^k} (\mathcal{H}(\mathcal{P} \cup \{\mathbf{y}\}) - \mathcal{H}(\mathcal{P})) \prod_{j=1}^m \phi_j[y_j] dy_j \\ &= \int_{l_1^k}^{u_1^k} \dots \int_{l_m^k}^{u_m^k} \sum_{k'=1}^q \prod_{j=1}^m (u_j^{k'} - \max\{l_j^{k'}, l_j^k\}) \phi_j[y_j] dy_j \\ &= \sum_{k'=1}^q \prod_{j=1}^m \int_{l_j^k}^{u_j^{k'}} (u_j^{k'} - \max\{l_j^{k'}, l_j^k\}) \phi_j[y_j] dy_j \\ &= \sum_{k'=1}^q \prod_{j=1}^m G(l_j^k, u_j^k, l_j^{k'}, u_j^{k'}). \end{aligned}$$

For each cell $k' = 1 \dots q$ contributions are calculated per dimension as follows,

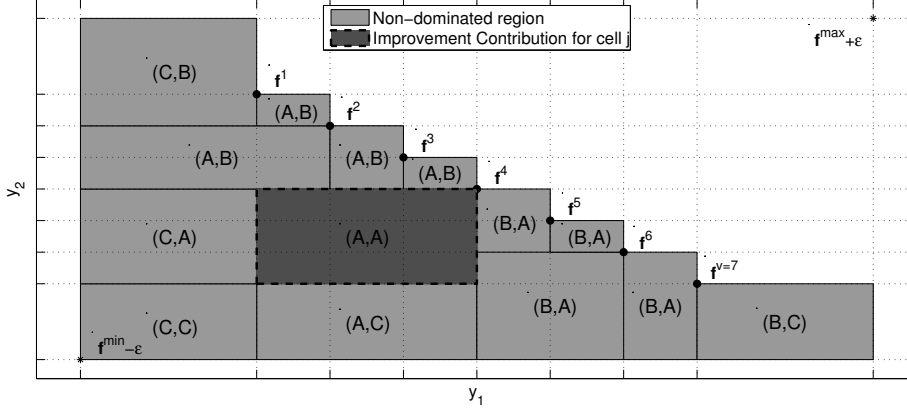


Fig. 3: The Improvement Contribution (IC) of the dark shaded cell k is calculated by multiplying different types of improvement (A, B or C) per dimension for all cells. The final IC is obtained by summation over all cells.

Type	$G(l_j, u_j, a_j, b_j) =$	Condition
A	$(b_j - a_j)(\Phi_j[\max(a_j, l_j)] - \Phi_j[l_j]) +$ $(b_j - \mu_j(\mathbf{x}))(\Phi_j[\min(b_j, u_j)] - \Phi_j[\max(a_j, l_j)]) +$ $s_j^2(\mathbf{x})(\phi_j[\min(b_j, u_j)] - \phi_j[\max(a_j, l_j)])$	$b_j > l_j \wedge a_j < u_j$
B	$(b_j - a_j)(\Phi_j[u_j] - \Phi_j[l_j])$	$a_j \geq u_j$
C	0	otherwise

There are several ways a cell k' can contribute to the IC_k depending on its position relative to cell k , see Figure 3. Cell k is represented by the dark shaded cell. The pairs (\cdot, \cdot) inside each cell $k' = 1 \dots q$ denote the type of contribution (A, B or C) per dimension. A, B and C refer to the Equations of the piecewise function G , see the previous Table. If a cell is completely covered by cell k in any one dimension, its length in that dimension is always included (type B). If a cell is only partially covered by the cell k , the contribution is divided into two parts: the integration when the cell is possibly fully covered (first line of A; zero if $a_j \leq l_j$) and when the cell is partially covered (second and third line of A). If a cell is not covered in any dimension, it also does not contribute to the IC_k (type C). The intermediary contribution IC_k is obtained by multiplying the different kinds of contributions for each dimension and summing it over all cells $k' = 1 \dots q$.

The correctness of this algorithm has been verified by an extensive numerical comparison against the publicly available code of Emmerich et al. [16] for the two objective case. In addition, the two and three objective cases have been verified using Monte Carlo methods.

Regardless of the fact that the new procedure is already significantly cheaper than the method proposed by Emmerich et al. [16], this hypervolume-based EI is still more expensive to evaluate than other statistical criteria. This is due to the computation time being more sensitive to the number of cells as well as the reliance on a binary

partitioning of the non-dominated region into disjoint cells, which requires more cells to cover the integration area than the Walking Fish Group (WFG) algorithm explained in section 3.4. Hence, a simplification of the hypervolume-based EI is proposed in the next section which can be evaluated using the WFG algorithm.

3.3.2 Hypervolume-based PoI

Inspired by the definition of the hypervolume-based EI [16], the hypervolume-based PoI can be written as the product of the improvement function $I(\mu, \mathcal{P})$ and the PoI $P[I]$, and so the advantages of using the hypervolume contribution can be preserved while significantly reducing the overall computational complexity,

$$P_{hv}[I] = I(\mu, \mathcal{P}) \cdot P[I], \quad (17)$$

where $I(\mu, \mathcal{P})$ is defined as in (14) and $\mu = (\mu_1(\mathbf{x}), \dots, \mu_m(\mathbf{x}))$ is a vector that contains the prediction of the Kriging models of each objective function for a point \mathbf{x} . In effect, the prediction variance is not taken into account anymore for the improvement function, in contrast to EI_{hv} , as it is moved outside of the integral.

The integration area A of $P[I]$ corresponds to the non-dominated region and, hence, a closed-form expression of the hypervolume-based PoI can be derived from the same set of cells used to evaluate $P[I]$, see Figure 2b, namely,

$$P_{hv}[I] = \left(\sum_{k=1}^q \pm Vol(\mu, \mathbf{l}^k, \mathbf{u}^k) \right) \cdot P[I] \quad (18)$$

where,

$$Vol(\mu, \mathbf{l}, \mathbf{u}) = \begin{cases} \prod_{j=1}^m (u_j - \max(l_j, \mu_j(\mathbf{x}))) & \text{if } u_j > \mu_j(\mathbf{x}) \text{ for } j = 1 \dots m \\ 0 & \text{otherwise} \end{cases}.$$

3.3.3 Euclidean distance-based improvement function

Similarly to the hypervolume-based PoI, Keane et al. [26] defines the EI as the product of the PoI $P[I]$ and an Euclidean distance-based improvement function. Let \mathbf{f}^c be the solution in \mathcal{P} that is located closest to the centroid $\hat{\mathbf{y}}(\mathbf{x})$ of the $P[I]$ integral,

$$\mathbf{f}^c = \underset{\mathbf{f}^c \in \mathcal{P}}{\operatorname{argmin}} \sqrt{\sum_{j=1}^m w_j (\hat{\mathbf{y}}_j(\mathbf{x}) - \mathbf{f}_j^c)^2}, \quad (19)$$

where the weight vector $\mathbf{w} = (w_1, \dots, w_m)$ is used to scale the objective functions into the same range and,

$$\hat{\mathbf{y}}_j(\mathbf{x}) = \sum_{k=1}^q \hat{y}_j(\mathbf{x}; \mathbf{l}^k, \mathbf{b}^k), \quad (20)$$

with the centroid $\hat{\mathbf{y}}$ over arbitrary integral bounds $[\mathbf{l}, \mathbf{b}]$ defined by,

$$\begin{aligned}
\hat{y}_j(\mathbf{x}; \mathbf{l}, \mathbf{b}) &= \int_{l_1^{u_1}} \dots \int_{l_m^{u_m}} \phi_1[y_1] \dots y_j \phi[y_j] \dots \phi_m[y_m] dy_m \dots dy_1 / P[I] \\
&= \prod_{j'=1}^{j-1} (\Phi_{j'}[u_{j'}] - \Phi_{j'}[l_{j'}]) \times \prod_{j'=j+1}^m (\Phi_{j'}[u_{j'}] - \Phi_{j'}[l_{j'}]) \\
&\quad \times (\mu_j(\mathbf{x})\Phi_j[u_j] - s_j^2(\mathbf{x})\phi_j[u_j] - \mu_j(\mathbf{x})\Phi_j[l_j] + s_j^2(\mathbf{x})\phi_j[l_j]) / P[I].
\end{aligned} \tag{21}$$

Lastly, $E_{euclid}[I]$ for an input vector \mathbf{x} is defined as,

$$E_{euclid}[I] = \sqrt{\sum_{j=1}^m w_j (\hat{\mathbf{y}}_j(\mathbf{x}) - \mathbf{f}_j^c)^2 \cdot P[I]}. \tag{22}$$

Like all other EI criteria the integration area A of $P[I]$ and $\hat{\mathbf{y}}(\mathbf{x})$ is the non-dominated region.

3.4 Decomposing the objective space into cells

In order to evaluate these statistical criteria efficiently, one or more integrals need to be evaluated over an integration area A . As A is non-rectangular and often irregularly shaped, especially for a higher number of objective functions, the integral must first be decomposed into a sum of k integrals over rectangular cells. While these cells can be identified analytically upfront for two objectives [26] or one can use the most fine-grained cells possible (for a total of $q \approx (v+1)^m$ cells; [16]), this becomes rather prohibitively complex and cumbersome for a higher number of objective functions (> 2).

Instead, the authors propose to decompose the integration area in as few cells as possible using an efficient computer algorithm, i.e., each cell encompasses a large part of the integration area. A straightforward approach to determine the required bounds of the cells for the evaluation of the criteria is to use binary partitioning [11], see Figure 2a. While this approach is quite flexible as it allows to identify different kind of integration areas (e.g., leading to several variants of statistical criteria), it becomes prohibitively expensive as the number of objectives exceeds four. Nonetheless, by terminating the binary partitioning early the statistical criteria can still be approximated fairly well for a higher number of objectives.

However, the focus of this work is to improve the performance of the exact evaluation of the criteria. To that end, it makes sense to take advantage of the numerous algorithms for calculating the hypervolume. Formally, the hypervolume is the Lebesgue integral,

$$\mathcal{H}(\mathcal{P}) = \int_{\mathbf{y} \in A} 1 dy_1 \dots dy_m, \tag{23}$$

where A is the region dominated by the Pareto set \mathcal{P} and bounded by some reference point \mathbf{r} . As the statistical criteria are integrals evaluated over a similar area, but using a different integrand, the idea is to adapt a hypervolume routine and retrieve the integration area A as a set of cells instead of immediately calculating its (hyper)volume.

Exact algorithms [31, 43, 4, 3] for calculating the hypervolume as well as approximations [28, 7, 8, 20, 1, 2] and alternative versions of the hypervolume problem, e.g., finding

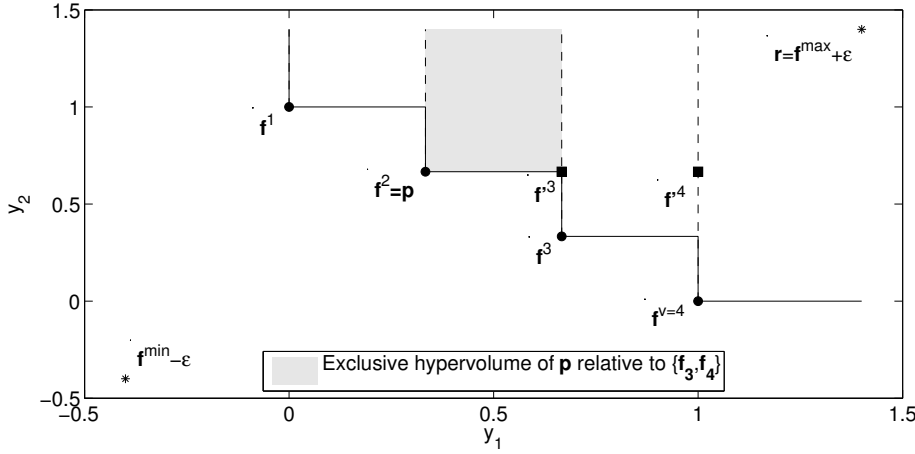


Fig. 4: One step in the WFG algorithm to calculate $\mathcal{H}_{exc}(\mathbf{f}^2 = \mathbf{p}; \{\mathbf{f}^3, \mathbf{f}^4\})$, denoted by the shaded region (\mathbf{f}^1 has already been processed). The exclusive hypervolume slice is efficiently calculated as the volume of the cell bounded by \mathbf{p} and \mathbf{r} minus the hypervolume of a reduced Pareto set, represented by the squares, where all points are limited by the contributing point \mathbf{p} . This creates many dominating points which can be removed before continuing calculation. In this illustration the volume of the cell bounded by \mathbf{f}_3' and \mathbf{r} is subtracted to calculate the exclusive hypervolume.

the Pareto point(s) that contributes least to the hypervolume [9], have been suggested in literature. While the algorithm proposed by Beume et al. [3] has the best worst case complexity, the Walking Fish Group (WFG) algorithm [39] is actually faster on most practical optimization problems and, hence, is adapted in this work to evaluate the statistical criteria.

The basic WFG algorithm operates by defining the hypervolume as a sum of exclusive hypervolumes, originally introduced by [18],

$$\mathcal{H}(\mathcal{P}) = \sum_{i=1}^v \mathcal{H}_{exc}(\mathbf{f}(\mathbf{x}_i^*); \mathcal{P} \setminus \{\mathbf{f}(\mathbf{x}_1^*), \dots, \mathbf{f}(\mathbf{x}_i^*)\}),$$

where each exclusive hypervolume in the summation corresponds to one of the slices bounded by dashed lines in Figure 4. At first sight this may look expensive as by definition the exclusive hypervolume (13) itself requires two separate hypervolume calculations. Fortunately, several optimizations can be made based on the following main ideas, see Figure 4.

- Slices to the left of the contributing point \mathbf{p} (i.e., with smaller values than \mathbf{p} in the current objective) can be discarded as they contain no hypervolume that is dominated by \mathbf{p} .
- If \mathbf{p} is dominated in the remaining (unprocessed) objectives it dominates no more exclusive hypervolume.

Advantage of these insights can be taken by rewriting the exclusive hypervolume from (13) as $\mathcal{H}_{exc}(\mathbf{p}; \mathcal{P}) = \mathcal{H}(\{\mathbf{p}\}) - \mathcal{H}(\mathcal{P}')$, where $\mathcal{P}' = \{\text{limit}(\mathbf{p}, \mathbf{q}) | \mathbf{q} \in \mathcal{P}\}$ and $\text{limit}(\mathbf{p}, \mathbf{q}) = (\max\{p_1, q_1\}, \dots, \max\{p_m, q_m\})$. The first term $\mathcal{H}(\{\mathbf{p}\})$ is simply the

volume of the cell bounded by \mathbf{p} and the reference point \mathbf{r} , the second term $\mathcal{H}(\mathcal{P}')$ is a recursion where the hypervolume is calculated for \mathcal{P}' . \mathcal{P}' is obtained by taking the maximum (*limit*) of the contributing point and for each point in \mathcal{P} , i.e., the points are projected on \mathbf{p} , see Figure 4. Obviously, many points will be introduced that are dominated by some other point in \mathcal{P}' . These points do not contribute anything to the hypervolume and can be removed by taking the non-dominated subset of \mathcal{P}' before continuing calculation. It is this last step that significantly improves the performance as it has been shown that most datasets already lose over 50% of their points after only one recursion.

Note that if the contributing point \mathbf{p} has generally higher objective values then likely more points can be pruned. Hence, it makes sense to process the worse Pareto points first by sorting the Pareto set \mathcal{P} descending on the current objective before each iteration. Moreover, now the hypervolume calculation can also be sliced on the sorted objective each recursion, hence the use of the terms current objective and unprocessed objectives in the previous paragraph. Each recursion the current objective is sorted and is not considered in subsequent calculations of \mathcal{H}_{excl} , for more information on this optimization please see [38]. This is similar to the Hypervolume by Slicing Objectives algorithm (HSO;[40]). A final optimization is the base case for two objectives, for which we can easily calculate the hypervolume in $O(v)$ assuming the Pareto set is sorted on the current objective. The base case for two objectives is similar to the algorithm proposed by [26] to calculate the Euclidean distance-based expected improvement and, hence, the performance of the WFG algorithm is at least equal to it.

The WFG algorithm is easily adapted to keep a record of a cell's lower- and upper-bound instead of calculating its hypervolume. The adapted WFG algorithm, in case of minimization, identifies the cells that are dominated by the Pareto set. Naturally, the algorithm can be modified to find the dominated region by temporarily viewing it as a maximization problem. However, for the evaluation of the statistical criteria, especially the EI, it is much more useful to identify the non-dominated region. This is achieved by subtracting the cells obtained from the adapted WFG algorithm from the cell that covers \mathbb{R}^m (possibly bounded by a reference point \mathbf{r} when using one of the hypervolume-based criteria), see Figure 2a.

Hence, the adapted WFG algorithm can be used for statistical criteria where the integration area A corresponds to either the dominated or non-dominated region. Should other integration areas be required, e.g., for calculating the probability that a new point dominates at least two Pareto points, more flexible but slower methods such as binary partitioning [11] can be used.

After $q \ll (v+1)^m$ sets of cells (=integral bounds) have been identified the actual PoI and EI statistical criteria can be evaluated using Equations (10), (20), (16) or (18). While evaluating the criteria the point $\mathbf{f}^{max} + \epsilon$ is replaced by (∞, \dots, ∞) or \mathbf{r} depending on the criterion.

A plot with the practical computation time and the number of the cells is shown in Figures 5a and 5b, applying the adapted WFG algorithm to sets of Pareto points randomly drawn from the first quadrant of a unit sphere (taking the mean values of 1000 repetitions). The computation time of the cells poses no problem for the evaluation of the criteria (well within the seconds range). The limited factor of the EMO algorithm is the number of cells the integration area A is decomposed into, as for each cell the corresponding PoI or EI equations needs to be evaluated which can become prohibitively expensive when many cells are required to identify the integration area A .

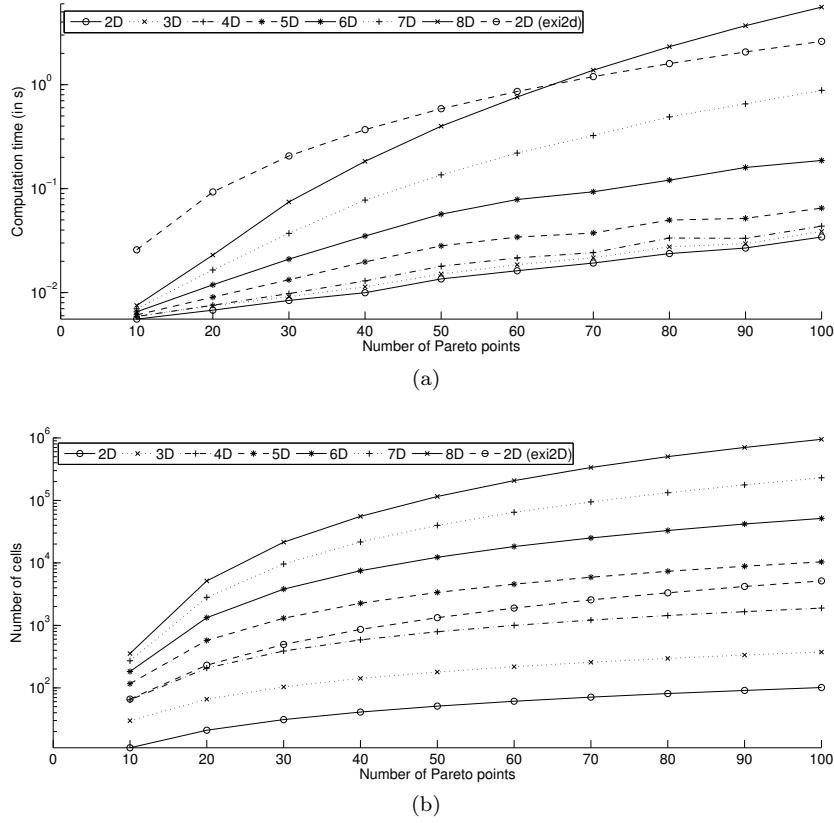


Fig. 5: a) Computation time of the integral bounds versus the number of Pareto points, for a different number of objective functions. b) The number of cells versus the number of Pareto points. The evaluation of the statistical criteria is limited by the number of cells the integration area A is decomposed into. The computation time and number of cells for one run of the exi2D algorithm [16] for an arbitrary input point \mathbf{x} is also included for the 2D case. In contrast to the proposed algorithm, the exi2D algorithm needs to be run multiple times during optimization of the criteria.

The decomposition of the integration area A into cells and the actual evaluation of the PoI and EI criteria is separate, in the sense that the cells only need to be identified once every sampling iteration, and then only if the (intermediate) Pareto set has changed with respect to the previous iteration. Afterwards, the PoI and EI criteria can be evaluated multiple times for a point \mathbf{x} , e.g., during optimization, using the same set of cells. Lastly, the hypervolume indicator is also easily obtained by summation of the volume of the cells.

Function	d	m	Reference point r
DTLZ1	6 inputs	3 objectives	(400, 400, 400)
DTLZ2	6 inputs	3 objectives	(2.5, 2.5, 2.5)
DTLZ7	6 inputs	4 objectives	(1, 1, 1, 50)
DTLZ5	6 inputs	6 objectives	(2.5, 2.5, 2.5, 2.5, 2.5, 2.5)

Table 1: Summary of the DTLZ benchmark functions.

4 Examples

4.1 Introduction

A good set of configurable multiobjective benchmark problems has been proposed by Deb et al. [15], of which four benchmark functions are chosen and adapted slightly to benchmark the EMO algorithm. A summary of the selected benchmark functions is found in Table 1. For a complete description of the benchmark functions the authors refer to [15].

All benchmark functions are configured to have six input parameters. Specifically, the first example is the DTLZ1 function with three objective functions where the Pareto front lies on the plane $y_1 + y_2 + y_3 = 1$. The second example is the DTLZ2 function with three objective functions where the Pareto front is the first quadrant of an unit sphere centered on the origin. The third example is the DTLZ7 function with four objective functions which has $2^{m-1} = 2^{4-1} = 8$ disconnected Pareto-optimal regions in the objective space. The last example, the DTLZ5 function configured to have six objective functions, is similar to DTLZ2 except that the Pareto front is just one slice of the unit hypersphere, i.e., the Pareto front is a (densely populated) curve in a $m = 6$ dimensional objective space.

4.2 Experimental setup

An initial set of 65 samples is generated by a near-optimal maximin Latin Hypercube Design (LHD; [12]). Subsequently, a statistical criterion is optimized for each iteration to select the next point to evaluate. The criterion is optimized using a combination of Monte Carlo sampling and a local search. Specifically, $20 \times n$ Monte Carlo candidate points are generated and evaluated on the criterion. The best Monte Carlo candidate is further refined using Matlab's `fmincon` optimizer.

Various configurations of the EMO algorithm are applied on the benchmark functions. In particular, EMO is configured with the EI_{euclid} and P_{hv} criterion together with Kriging models using the Matérn correlation function [32] with $\nu = \frac{3}{2}$ and a constant regression function ($M = 1$ and $F = 1$). The hyperparameters of the Kriging models are optimized using SQPLab [6] (<http://www-rocq.inria.fr/~gilbert/modulopt/optimization-routines/sqplab/sqplab.html>), utilizing likelihood derivative information. The optional weight vector \mathbf{w} of the EI_{euclid} criterion is set to $\mathbf{1}$ for all benchmark functions, except for DTLZ7 where $\mathbf{w} = (1, 1, 1, 0.02)$ to scale the last objective function into the same range as the other objective functions.

Furthermore, the EMO runs of the EI_{euclid} criterion are repeated with Kriging models using the Gaussian correlation function, these runs are denoted by EI_{euclid}^{gauss} in

the results. Lastly, extra EMO runs are configured for the DTLZ1 and DTLZ2 functions using the expensive EI_{hv} criterion with Kriging models using the Matérn correlation. The EI_{hv} criterion is not applied to the DTLZ5 and DTLZ7 problems. Each of the in total 14 EMO runs is repeated 10 times for statistical robustness and halts when the sample budget is met, namely, 250 samples.

The EMO runs are compared against the NSGA-II, SPEA2 and SMS-EMOA evolutionary algorithms with a varying population size and maximum number of generations. The first run is configured with a population size of 25 and a maximum number of generations of 10 (total sample budget 250) and the second run is configured with a population size of 50 and a maximum number of generations of 50 (total sample budget 2500). The remaining parameters have been left to their default values. Similarly to the EMO runs, the evolutionary algorithm runs are repeated 10 times.

Beside assessing the performance of the algorithms using the hypervolume metric, the convergence measure is used too. The convergence measure is the mean distance of every point of the Pareto set to the closest Pareto point of the known Pareto front. In this work the known Pareto fronts are represented by 100000 Monte Carlo points.

4.3 Results

Results for the benchmark functions have been summarized in Table 2. Note that the differences on the hypervolume metric are more significant than they appear because of the conservative choice of the reference point \mathbf{r} (needed to accommodate the results of all test configurations).

In general, it is seen that the EMO runs have better performance than the MOEAs in terms of hypervolume score for most functions except for DTLZ1. After a closer examination it is observed that the accuracy of the Kriging models of DTLZ1 for most statistical criteria is sub-optimal. In particular, the first objective function is difficult to approximate using the Kriging models, see Figure 6a.

A plot of the final Pareto sets generated of the DTLZ2 problem is shown in Figure 7. It is seen that the hypervolume-based criteria emphasizes the edges of Pareto front more while leaving a small gap between the edge and the inner portion of the Pareto front. This is not unlike the DTLZ2 results as reported in [5] and is due to the nature of the hypervolume indicator. Logically, the farther away the reference point is located, the larger the exclusive hypervolume will be for points lying on the edge of the current Pareto set (as the exclusive hypervolume is then solely bounded by the reference point). Further research is needed to determine the influence of the choice of reference point \mathbf{r} on the statistical criteria [1].

While the EMO algorithm outperforms the MOEAs on the hypervolume indicator on most problems, there are some limitations. The EMO algorithm, and other MOSBO techniques, rely on the quality of the surrogate model to guide the selection of new expensive data points. The Kriging models do not have to be accurate at the start of the algorithm when using the EI and PoI criteria, but they should be able to capture the behavior of the objective functions sufficiently well when enough samples become available, which might not always be the case (see Figure 6 and the DTLZ1 results). Furthermore, the construction of the Kriging models and the evaluation of the statistical criteria comes at a computational cost, similar to the computational cost of MOEAs that rely on the hypervolume (i.e., SMS-EMOA), which might limit the practical usage of the EMO algorithm for some (less expensive) optimization problems.

Problem	X	Algorithm	Convergence measure		Hypervolume	
			Mean	Std	Mean	Std
DTLZ1	250	EI_{euclid}	93.2833	18.7840	6.3498e7	2.4970e5
		EI_{euclid}^{gauss}	100.6741	14.2258	6.3650e7	1.2418e5
		EI_{hv}	37.6112	2.9315	<i>6.3940e7</i>	6.0452e4
		P_{hv}	66.9199	14.0029	6.3838e7	7.4330e4
		NSGA-II	75.8391	20.4219	6.3612e7	2.3441e5
		SPEA2	104.6259	0	6.3482e7	0
		SMS-EMOA	44.8818	7.9740	6.3976e7	8.0982e3
	2500	NSGA-II	16.6888	4.8071	6.3991e7	1.0227e4
		SPEA2	93.8381	0	6.3984e7	0
		SMS-EMOA	9.5047	2.8750	6.4000e7	324.0575
DTLZ2	250	EI_{euclid}	0.0843	0.0205	14.9423	0.0181
		EI_{euclid}^{gauss}	0.1481	0.0133	14.8994	0.0114
		EI_{hv}	0.0411	0.0052	14.8834	0.0165
		P_{hv}	0.0106	0.0021	15.0326	0.0054
		NSGA-II	0.2725	0.0460	13.6238	0.2725
		SPEA2	0.1643	0	14.4873	0
		SMS-EMOA	0.0388	0.0071	14.9021	0.0160
	2500	NSGA-II	0.1497	0.0185	14.6435	0.0460
		SPEA2	0.1544	0.0298	14.8503	0
		SMS-EMOA	0.0030	2.8954e-4	15.0280	3.4727e-4
DTLZ7	250	EI_{euclid}	4.3888	2.8159	42.4629	0.4042
		EI_{euclid}^{gauss}	1.7066	1.4069	42.6332	0.3295
		P_{hv}	0.0280	0.0037	43.5404	0.0188
		NSGA-II	13.9371	2.3112	23.2392	5.4733
		SPEA2	10.1169	0	37.4830	0
		SMS-EMOA	3.4186	2.2457	41.2087	1.6529
	2500	NSGA-II	9.6799	2.3516	30.7966	4.2005
		SPEA2	5.4330	0	42.1191	0
		SMS-EMOA	0.0236	0.0015	43.7127	0.0953
DTLZ5	250	EI_{euclid}	0.2259	0.0019	197.1390	0.1453
		EI_{euclid}^{gauss}	0.2286	0.0013	196.8852	0.1777
		P_{hv}	<i>0.0835</i>	0.0053	198.6425	0.1563
		NSGA-II	0.0656	0.0376	192.1285	2.0064
		SPEA2	0.1475	0	192.6617	0
		SMS-EMOA	0.0467	0.0268	196.0038	0.6004
	2500	NSGA-II	0.0727	0.0162	194.9017	0.3805
		SPEA2	0.2151	0	194.3750	0
		SMS-EMOA	0.1141	0.0070	198.5351	0.0343

Table 2: Results of the EMO algorithm, NSGA-II, SPEA2 and SMS-EMOA. The best results for each test function are highlighted in bold, for each performance metric and within the same sample budget. The best results among the different configurations of the EMO algorithm are marked as italic.

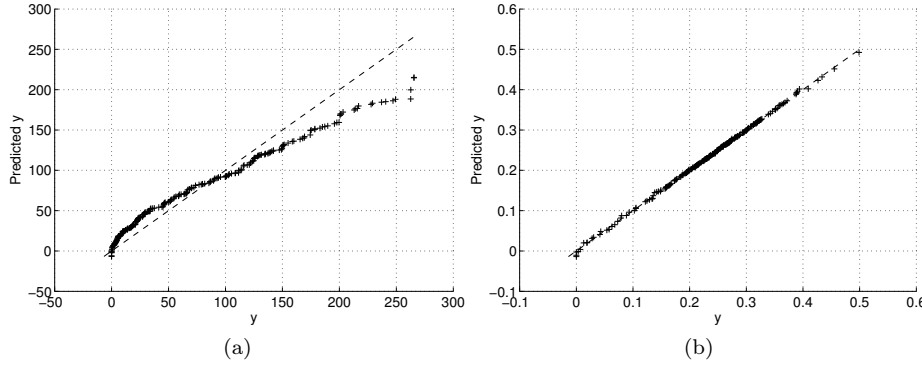


Fig. 6: 20-fold cross validation applied on the Kriging models based on 250 samples. The black dots denote the cross validated prediction values versus the real objective values. a) Final Kriging model of the first objective function of the DTLZ1 function. It is seen that Kriging has problems approximating the larger values of the objective function. b) Final Kriging model of the first objective function of the DTLZ5 function. Kriging is able to approximate the objective function quite well.

Specifically, the construction of the Kriging models and the thorough optimization of the statistical criteria make the EMO algorithm more expensive than SMS-EMOA. Moreover, the EMO algorithm requires the storage of the integral bounds in memory, which can be prohibitively expensive for a higher number of objectives.

5 Conclusion

The authors presented the Efficient Multiobjective Optimization (EMO) algorithm, which uses multiobjective versions of the Probability of Improvement (PoI) and Expected Improvement (EI) to identify the Pareto front with a limited sample budget. Different configurations of the EMO algorithm are compared against the well-known SPEA2, NSGA-II and SMS-EMOA evolutionary methods with promising results. In theory an arbitrary number of objective functions can be handled. However, in practice due to the nature of the multiobjective EI and PoI statistical criteria the EMO algorithm also does not escape the curse of dimensionality (no-free-lunch theorem) with respect to the number of objective functions and number of Pareto points. Nevertheless, the EMO algorithm can be applied to problems up to eight objectives. Furthermore, it should be noted that in practice a high number of objectives are often aggregated to a smaller number of objectives (≤ 4) using, e.g., weighted sums, as such high-dimensional spaces are not easily interpretable for designers.

Future work will focus more on exploring the key benefits of the EMO algorithm on various industrial applications and benchmark problems. Preliminary results show that the benefits of the EMO algorithm are more pronounced when limited to a very low number of function evaluations (< 100). Moreover, the benefits of using statistical criteria, which balance exploration and exploitation automatically, may manifest more for problems with a complex optimization landscape and multiple local Pareto fronts. In addition, future work will focus on minimizing the number of cells and on an iterative

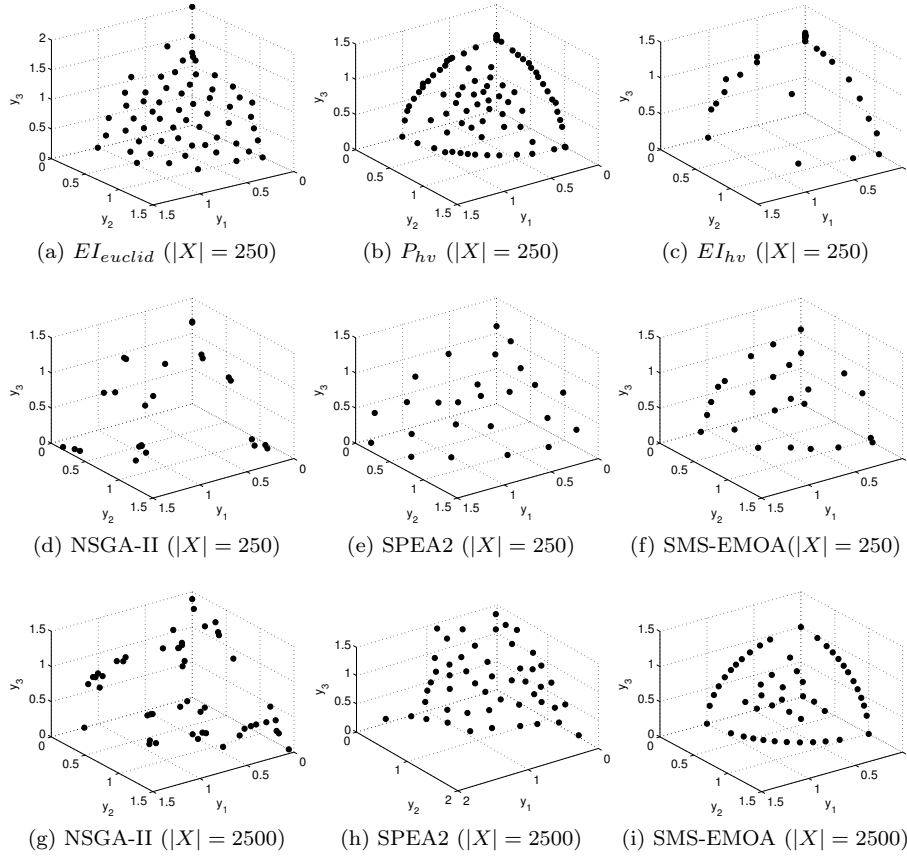


Fig. 7: Generated Pareto sets of the DTLZ2 function. The hypervolume-based metric focuses more on sampling the edge (extrema) of the Pareto front, while the Euclidean distance-based criterion performs a seemingly more uniform search over the Pareto front, though it performs slightly worse on the hypervolume metric.

update scheme for the cells, which will be considerable more efficient than recalculating the cells almost each iteration. Indirectly, a speedup can also be achieved by selecting multiple update points at a time. Finally, it may also be worthwhile to investigate the use of approximated statistical criteria, namely, adapting hypervolume approximation routines for decomposing the integration area into cells.

Acknowledgments

This work was supported by the Fund for Scientific Research in Flanders (FWO-Vlaanderen). Ivo Couckuyt and Dirk Deschrijver are post-doctoral research fellows of FWO-Vlaanderen. This research has (partially) been funded by the Interuniversity Attraction Poles Program BESTCOM initiated by the Belgian Science Policy Office.

References

1. A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: Optimal μ -distributions and the choice of the reference point. In *Workshop Foundation Genetic Algorithms*, 2009.
2. J. Bader, K. Deb, and E. Zitzler. Faster hypervolume-based search using monte carlo sampling. In *Multiple Criteria Decision Making Sustainable Energy Transportation System*, 2010.
3. N. Beume. S-metric calculation by considering dominated hypervolume as klee's measure problem. *Evolutionary Computation*, 17(4):477–492, 2009.
4. N. Beume, C. M. Fonseca, and M. López-Ibáñez. On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082, 2009.
5. N. Beume, B. Naujoks, and M.T.M. Emmerich. Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.
6. J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C.A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer, 2006.
7. K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: Np-hard in general, but fast in practice. In *Evolutionary Multiobjective Optimization*, 2009.
8. K. Bringmann and T. Friedrich. Don't be greedy when calculating hypervolume contributions. In *Workshop Foundation Genetic Algorithms*, 2009.
9. K. Bringmann and T. Friedrich. An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, 18(3):383–402, 2010.
10. I. Couckuyt, F. Declercq, T. Dhaene, and H. Rogier. Surrogate-based infill optimization applied to electromagnetic problems. *Journal of RF and Microwave Computer-Aided Engineering: Advances in design optimization of microwave/rf circuits and systems*, 20(5):492–501, 2010.
11. I. Couckuyt, D. Deschrijver, and T. Dhaene. Towards efficient multiobjective optimization: Multiobjective statistical criterions. In *IEEE World Congress on Computational Intelligence*, pages 1–7, 2012.
12. E.R. Dam, B.G.M. van Husslage, D. den Hertog, and J.B.M. Melissen. Maximin Latin hypercube designs in two dimensions. *Operations Research*, 55(1):158–169, 2007.
13. Jos de Brabanter. *LS-SVM Regression Modelling and its Applications*. PhD thesis, Katholieke Universiteit Leuven, 2004.
14. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.
15. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
16. M.T.M. Emmerich, A.H. Deutz, and J.W. Klinkenberg. Hypervolume-based expected improvement: Monotonicity properties and exact computation. In *IEEE Congress on Evolutionary Computation (CEC)*, 2011.
17. M.T.M. Emmerich, K. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.
18. M. Fleischer. The measure of pareto optima: Applications to multi-objective metaheuristics. In *Evolutionary Multi-Criterion Optimization*, pages 519–533, 2003.
19. A. Forrester, A. Sobester, and A. Keane. *Engineering Design Via Surrogate Modelling: A Practical Guide*. Wiley, Chichester, 2008.
20. T. Friedrich, C. Horoba, and F. Neumann. Multiplicative approximations and the hypervolume indicator. In *Genetic Evolutionary Computation*, 2009.
21. A. Gaspar-Cunha and A. Vieira. A multi-objective evolutionary algorithm using neural networks to approximate fitness evaluations. *International Journal of Computers, Systems, and Signals*, 6(1):18–36, 2004.
22. D. Gorissen, K. Crombecq, I. Couckuyt, P. Demeester, and T. Dhaene. A surrogate modeling and adaptive sampling toolbox for computer based design. *Journal of Machine Learning Research*, 11:2051–2055, 2010.
23. S. Jeong and S. Obayashi. Efficient global optimization (ego) for multi-objective problem and data mining. In *Congress on Evolutionary Computation*, 2010.

24. D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Global Optimization*, 21:345–383, 2001.
25. D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, 1998.
26. A. J. Keane. Statistical improvement criteria for use in multiobjective design optimization. *AIAA Journal*, 44(4):879–891, 2006.
27. J. P.C. Kleijnen. *DASE : Design and Analysis of Simulation Experiments*. Springer, May 2007.
28. J. Knowles. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.
29. J. Knowles, D. Corne, and A. Reynolds. Noisy multiobjective optimization on a budget of 250 evaluations. In *5th International Conference on Evolutionary Multi-Criterion Optimization*, 2009.
30. J. Knowles and H. Nakayama. Meta-modeling in multiobjective optimization. In *Multiobjective Optimization: Interactive and Evolutionary Approaches*, pages 245–284. Springer-Verlag, Berlin, Heidelberg, 2008.
31. J. D. Knowles. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, University of Reading, UK, 2002.
32. C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
33. T.J. Santner, B.J. Williams, and W.I. Notz. *The design and analysis of computer experiments*. Springer series in statistics. Springer-Verlag, New York, 2003.
34. M.L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, 1999.
35. I. Voutchkov and A.J. Keane. Multiobjective Optimization using Surrogates. In I.C. Parmee, editor, *Adaptive Computing in Design and Manufacture 2006. Proceedings of the Seventh International Conference*, pages 167–175, Bristol, UK, April 2006.
36. T. Wagner, M. Emmerich, A. Deutz, and W. Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. *Parallel Problem Solving From Nature*, 6238:718–727, 2010.
37. G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–380, 2007.
38. L. While and L. Bradstreet. Applying the wfg algorithm to calculate incremental hypervolumes. In *IEEE World Congress on Computational Intelligence (WCCI)*, 2012.
39. L. While, L. Bradstreet, and L. Barone. A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16:86–95, 2012.
40. L. While, P. Hingston, L. Barone, and S. Huband. A faster algorithm for calculating hypervolume. *Evolutionary Computation*, 10(1):29–38, 2006.
41. Q. Zhang, W. Liu, E. Tsang, and B. Virginas. Expensive multiobjective optimization by moea/d with gaussian process model. *Evolutionary Computation*, 14(3):456–474, 2010.
42. A. Zhou, B-Y Qu, H. Li, S-Z Zhao, P. N. Suganthan, and Q. Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1:32–49, 2011.
43. E. Zitzler. Hypervolume metric calculation. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, 2001.
44. E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the performance of the strength pareto evolutionary algorithm. Technical report, Swiss Federal Institute of Technology, 2001.
45. E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V. Grunert da Fonseca. Performance assesment of multiobjective optimizers: an analysis and review. *Evolutionary Computation*, 7(2):117–132, 2003.